

In the Claims:

Please amend Claims 1, 15 and 30, all as shown below. Applicant respectfully reserves the right to prosecute any originally presented claims in a continuing or future application.

1. (Currently Amended) A method for dynamically binding a user interface to information, comprising:

presenting a user interface to a user wherein the user interface is operable to present information stored in a data source on a business object to the user, collect user additional information from the user, and store the user additional information in the data source on the business object;

defining a data binding tag that defines a boundary for rendering the information and rules for when the information is rendered, wherein the data binding tag includes a plurality of attributes;

specifying a first action by the data binding tag wherein the first action includes reading or updating the information stored in the first data source;

specifying, using a scripting language, at least one attribute on the data binding tag to reference the first data source associated with the first action, wherein the first data source is in the business object; and

rendering each item in the first data source in the user interface with a markup language according to the boundary and the rules defined by the data binding tag and based at least partially on the first action; and

wherein the first action can set or get the first data source.

2. (Previously Presented) The method of claim 1 wherein:

the data binding tag allows for the specification of JavaServer Page action elements.

3. (Previously Presented) The method of claim 1 wherein:

the scripting language is based on the Javascript language.

4. (Original) The method of claim 1 wherein:

the first data source identifies one of: 1) an object field; 2) an object property; and 3) an Extensible Markup Language document element.

5. (Original) The method of claim 4 wherein:  
an object is a JavaBean.
6. (Original) The method of claim 1 wherein:  
the first data source is one of: 1) an array; 2) a list; 3) a map.
7. (Previously Presented) The method of claim 1 wherein:  
the markup language can include at least one of: Hypertext Markup Language (HTML), Dynamic HTML, Extensible HTML, and Extensible Markup Language.
8. (Original) The method of claim 1 wherein:  
the first action can be a child of another action.
9. (Original) The method of claim 1 wherein:  
the first action can have at least one child action.
10. (Original) The method of claim 9 wherein:  
the at least one child action can have at least one other child action.
11. (Original) The method of claim 9 wherein:  
the at least one child action can selectively process the first data source.
12. (Original) The method of claim 9 wherein:  
the at least one child action can refer to the first data source with a context defined by the first action.
13. (Original) The method of claim 9 wherein:  
the at least one child action can perform at least one of the following actions on the first data source: 1) set; 2) get; 3) sort; and 4) filter.
14. (Original) The method of claim 9, further comprising:

rendering a list or a table based on the first data source.

15. (Currently Amended) A machine readable medium having instructions stored thereon that when executed by a processor cause a system to:

present a user interface to a user wherein the user interface is operable to present information stored in a data source on a business object to the user, collect user additional information from the user, and store the user additional information in the data source on the business object;

define a data binding tag that defines a boundary for rendering the information and rules for when the information is rendered, wherein the data binding tag specifies a first action and wherein the data binding tag includes a plurality of attributes and wherein the first action includes reading or updating the information stored in the first data source;

specify, using a scripting language, at least one attribute on the data binding tag to reference a first data source associated with the first action, ~~wherein the first data source is in the business object; and~~

render each item in the first data source in the user interface with a markup language according to the boundary and the rules defined by the data binding tag and based at least partially on the first action; and

~~wherein the first action can set or get the first data source.~~

16. (Previously Presented) The machine readable medium of claim 15 wherein:  
the data binding tag allows for the specification of JavaServer Page action elements.

17. (Previously Presented) The machine readable medium of claim 15 wherein:  
the scripting language is based on the Javascript language.

18. (Original) The machine readable medium of claim 15 wherein:  
the first data source identifies one of: 1) an object field; 2) an object property; and 3) an Extensible Markup Language document element.

19. (Original) The machine readable medium of claim 18 wherein:  
an object is a JavaBean.

20. (Original) The machine readable medium of claim 15 wherein:  
the first data source is one of: 1) an array; 2) a list; 3) a map.
21. (Previously Presented) The machine readable medium of claim 15 wherein:  
the markup language can include at least one of: Hypertext Markup Language (HTML),  
Dynamic HTML, Extensible HTML, and Extensible Markup Language.
22. (Original) The machine readable medium of claim 15 wherein:  
the first action can be a child of another action.
23. (Original) The machine readable medium of claim 15 wherein:  
the first action can have at least one child action.
24. (Original) The machine readable medium of claim 23 wherein:  
the at least one child action can have at least one other child action.
25. (Original) The machine readable medium of claim 23 wherein:  
the at least one child action can selectively process the first data source.
26. (Original) The machine readable medium of claim 23 wherein:  
the at least one child action can refer to the first data source with a context defined by  
the first action.
27. (Original) The machine readable medium of claim 23 wherein:  
the at least one child action can perform at least one of the following actions on the first  
data source: 1) set; 2) get; 3) sort; and 4) filter.
28. (Original) The machine readable medium of claim 23, further comprising instructions  
that when executed cause the system to:  
render a list or a table based on the first data source.

29. (Canceled)

30. (Currently Amended) A software framework for rendering at least one object on a user interface, comprising:

a set of data binding tags specified in a JSP programming language, wherein the data binding tags define a boundary and rules for when data is rendered and include a plurality of attributes to reference and display data, and wherein each tag can be used to bind and submit data that a user may edit in a web page;

wherein each attribute can include an expression, written in an expression language, that references a first business object that includes data collected from or presented to the user; and

wherein the tags specify a first action to be performed on the first business object wherein the first action includes reading or updating the information stored in the first data source; and

a markup language capable of rendering each item the data in the first business object in a user interface that is referenced by the expression according to the boundary and the rules defined by the data binding tag, wherein the expression is evaluated at rendering.

31. (Previously Presented) The framework of claim 30 wherein:

the data binding tag allows for the specification of JavaServer Page action elements.

32. (Previously Presented) The framework of claim 30 wherein:

the scripting language is based on the Javascript language.

33. (Original) The framework of claim 30 wherein:

the first data source identifies one of: 4) an object field; 2) an object property; and 3) an Extensible Markup Language document element.

34. (Original) The framework of claim 33 wherein:

an object is a JavaBean.

35. (Original) The framework of claim 30 wherein:

the first data source is one of: 4) an array; 2) a list; 3) a map.

36. (Previously Presented) The framework of claim 30 wherein:  
the markup language can include at least one of: Hypertext Markup Language (HTML), Dynamic HTML, Extensible HTML, and Extensible Markup Language.
37. (Original) The framework of claim 30 wherein:  
the first action can be a child of another action.
38. (Original) The framework of claim 30 wherein:  
the first action can have at least one child action.
39. (Original) The framework of claim 38 wherein:  
the at least one child action can have at least one other child action.
40. (Original) The framework of claim 38 wherein:  
the at least one child action can selectively process the first data source.
41. (Original) The framework of claim 38 wherein:  
the at least one child action can refer to the first data source with a context defined by the first action.
42. (Original) The framework of claim 38 wherein:  
the at least one child action can perform at least one of the following actions on the first data source: 4) set; 2) get; 3) sort; and 4) filter.
43. (Original) The framework of claim 38 wherein:  
a list or a table can be rendered based on the first data source.
44. (Previously Presented) A system for dynamically binding a user interface to information, comprising:  
a computer including a computer readable medium and processor operating thereon;

a user interface that is operable to

present information stored in a business object to a user,

collect ~~user~~ additional information from the user, and

store the ~~user~~ additional information within the business object;

a plurality of data binding tags written in a first programming language stored on the computer readable medium wherein each data binding tag

defines a boundary for rendering the information and rules for when the information is rendered,

includes a lifecycle associated therewith,

includes a plurality of attributes, and

specifies an action to be performed on the business object wherein the action includes ~~setting or getting reading or updating~~ information from the business object including the ~~user~~ additional information;

an expression language that can be used to evaluate expressions on specified business objects, wherein each expression specifies a business object in which the expression is to be evaluated; and

a markup language that can be used to render the specified business objects in the user interface according to the render boundary and the rules defined by the data binding tag.